| L Number | Hits | Search Text | DB | Time stamp |
|---|---|---|---|---|
| 1 | 9 | (("5721956") or ("5754888") or ("5787466") or ("5805787") or ("5890207") or ("5911779") or ("5933853") or ("6101576") or ("6389510")).PN. | USPAT; US-PGPUB; EPO; JPO; IBM_TDB | 2004/07/15 12:11 |
| 2 | 6 | ((("5721956") or ("5754888") or ("5787466") or ("5805787") or ("5890207") or ("5911779") or ("5933853") or ("6101576") or ("6389510")).PN.) and parallel | USPAT; US-PGPUB; EPO; JPO; IBM_TDB | 2004/07/15 12:11 |

US-PAT-NO:              5754888

DOCUMENT-IDENTIFIER:    US 5754888 A

TITLE:                  System for destaging data during
idle time by
                        transferring to destage buffer,
marking segment blank ,
                        reodering data in buffer, and
transferring to beginning
                        of segment


---------- KWIC ---------


US Patent No. - PN (1):
   5754888


Brief Summary Text - BSTX (7):
   Because of the mechanical nature of magnetic disks, the
performance of disks
has increased only gradually in the past.  One of the most
important
architectural advances in disks is RAID (Redundant Array of
Inexpensive Disks)
architecture pioneered by a group of researchers in UC
Berkeley, Katz, R. H.;
Gibson, A; and Patterson, D. A, Disk System Architectures
for High Performance
Computing, Proceeding of the IEEE, pp.  1842-1858, 1989.
The main idea of RAID
is using multiple disks in parallel to increase the total
I/O bandwidth which
scales with the number of disks.  Multiple disks in a RAID
can service a single
logical I/O request or support multiple independent I/Os in
parallel.  Since
the size and the cost of disks drop rapidly, RAID is a cost
effective approach
to high I/O performance.  One critical limitation of RAID
architecture is that
their throughput is penalized by a factor of four over
nonredundant arrays for
small writes which are substantial and are becoming a

dominant portion of I/O
workload.  The penalty results from parity calculations for
a new data, which
involves readings of old data and parity, and writings of
new data and parity.
A solution was proposed to the small-write problem by means
of parity logging,
Stodolsky, D.; Holland, M.; Courtright II, W. V.; and
Gibson, G. A., Parity
Logging Disk Arrays, ACM Transaction of Computer Systems,
pp.  206-235, 1994.
It was shown that with minimum overhead, parity logging
eliminates performance
penalty caused by RAID architectures for small writes.


Brief Summary Text - BSTX (14):
   Several techniques have been reported in the literature
in minimizing small
write costs in RAID systems.  Parity logging, an elegant
mechanism proposed by
utilizing the high transfer rate of large sequential data
to minimize small
write penalty in RAID systems, Stodolsky, D.; Holland, M.;
Courtright II, W.
V.; and Gibson, G. A., Parity Logging Disk Arrays, ACM
Transaction of Computer
Systems, pp.  206-235, August 1994.  They have shown that
with minimum
overhead, parity logging eliminates performance penalty
caused by RAID
architectures for small writes.  It was proposed a very
interesting approach
called write-twice to reduce the small write penalty of
mirror disks, Solworth,
J. A. and Orji, C. U., Distorted Mirrors, Proceedings at
the First
International Conference on Parallel and Distributed
Information Systems, pp.
10-17, 1991.  In their method several tracks in every disk
cylinder are
reserved.  When a write request comes, it is immediately
written to a closest
empty location, and the controller acknowledges the write
as complete.  Later
the data is written again to its fixed location.  Up to 80%
improvement in
small performance was reported.  It can also be used to

reduce write response
time in normal disks.  The write-twice method is normally implemented in the
disk controller level since it needs detailed timing information of disk drive.
It also requires substantial amount of disk storage to reserve tracks in each
cylinder.  Except for a few high-end products, most disk drives now use 2 or 3
platters per drive, implying only 4 to 6 tracks per cylinder.  Therefore, the
write-twice approach is mainly for those applications where cost is not the
primary concern.